

# DEVOPS-DAY-1

## SDLC

The **Software Development Life Cycle (SDLC)** is a process used by software developers to design, develop, test, and deploy software. It ensures that software is built in a systematic and efficient way. The SDLC consists of several phases, each with its specific activities.

Here are the basic phases of SDLC:

### **Requirement Gathering and Analysis:**

1. In this phase, the requirements for the software are collected from stakeholders (such as clients, users, etc.).
2. The goal is to understand what the software needs to do and what features are required.

### **System Design:**

1. Based on the requirements, a system design is created. This can include both high-level design (architecture) and detailed design (specific features and components).
2. It focuses on how the software will be structured and what technologies will be used.

### **Implementation (Coding):**

1. In this phase, the actual coding takes place. Developers write the code according to the design specifications.
2. The goal is to create a working version of the software.

### **Testing:**

1. After the software is developed, it goes through testing to identify and fix bugs or issues.
2. This includes various types of testing (unit testing, integration testing, system testing, etc.) to ensure the software works correctly.

### **Deployment:**

1. Once the software passes testing, it is deployed to the production environment, where users can start using it.
2. The software is released, and users begin to interact with it.

### **Maintenance:**

1. After deployment, the software enters the maintenance phase.
2. This involves updating the software for bug fixes, adding new features, or improving performance as needed



## **DEVOPS:**

DevOps is a way of working where development (Dev) and operations (Ops) teams work together closely to build, test, and release software faster and more reliably.

Key ideas of DevOps:

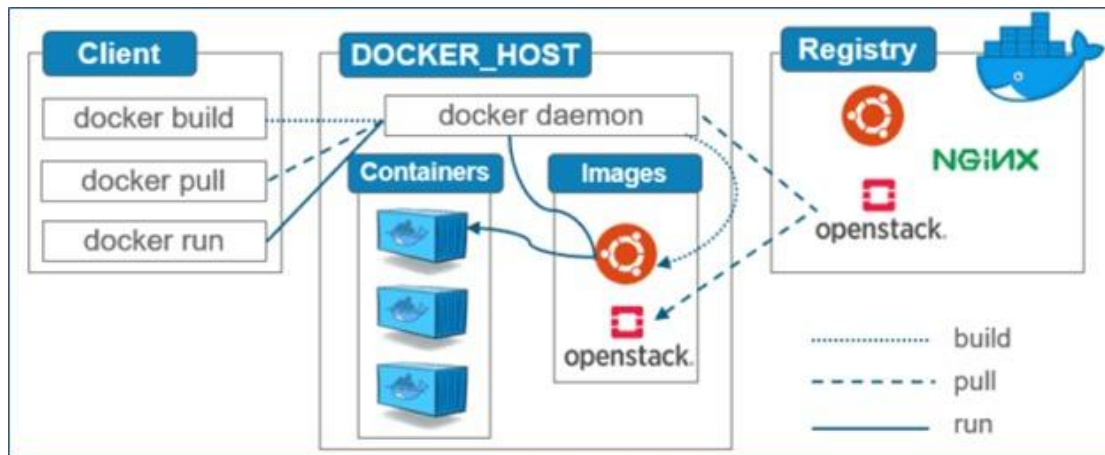
**Collaboration:** Developers and operations teams work together, instead of separately, to improve software.

**Automation:** Using tools to automatically test, build, and deploy software, saving time and reducing mistakes.

**Continuous Integration (CI):** Developers frequently update code, making sure it works well with the existing code by testing it automatically.

**Continuous Delivery (CD):** Software is always ready to be released, allowing for quicker updates and fixes.

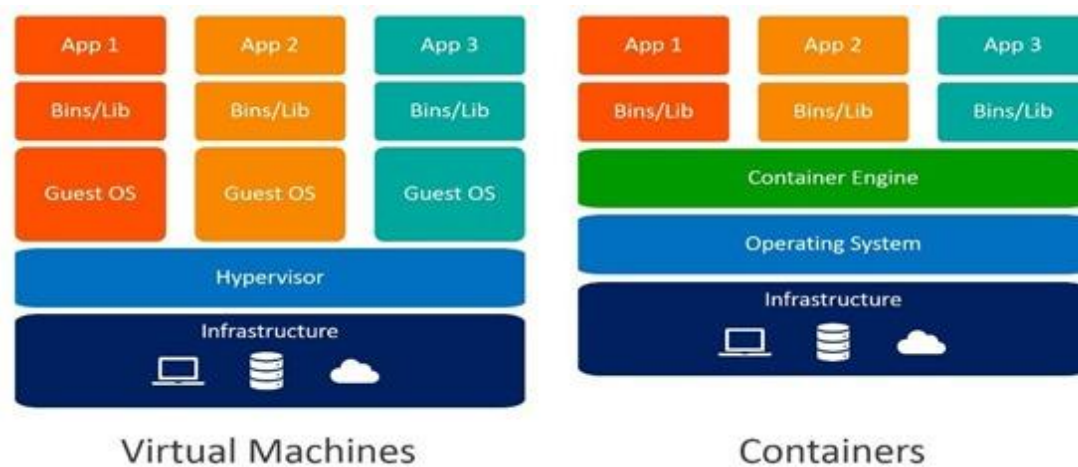
**Monitoring:** The software is constantly monitored to fix issues quickly and improve the product.



## VIRTUALIZATION:

Virtualization is the process of sharing hardware resources across several virtually isolated and mutually independent systems. It is achieved by using a hypervisor which acts as a bridge between the Operating System of each of the virtual machines and the underlying hardware.

Applications in virtual environments run on a host operating system on top of the hypervisor.



## DOCKER:

Docker is a platform that provides virtual containers on which an application can be deployed independent of the underlying OS of the server. Further the container can be created from a replica called docker image which contains all the dependencies and can run on any OS that has docker engine, with similar results.

# BASIC DOCKER COMMANDS

Display docker images available in our machine

```
$ docker images
```

Download docker image.

```
$ docker pull <image-name / image-id>
```

Run docker image.

```
$ docker run <image-name / image-id>
```

Delete docker image.

```
$ docker rmi <image-name / image-id>
```

Display all running docker containers.

```
$ docker ps
```

Display all running and stopped containers.

```
$ docker ps -a
```

Delete docker container.

```
$ docker rm <container-id>
```

Delete docker image forcefully.

```
$ docker rmi -f <image-id>
```

Stop Docker container.

```
$ docker stop <container-id>
```

